

Sub AS CLAIMS:

1. A computer-implemented method for automatically invoking a predetermined debugger command at a desired location of a single thread of a program containing at least one thread, said method comprising:

(a) embedding within said single thread at said desired location thereof a utility which reads a trace file in which said predetermined debugger command has been previously embedded; and

(b) running the program for reading said trace file and invoking said predetermined debugger command.

2. The method according to Claim 35, wherein said utility performs the following steps:

i. checks whether a trace file exists,

ii. if the trace file does not exist, creates the trace file and writes thereto a traced value of at least one variable at said desired location of the program, and

iii. if the trace file does exist, compares a current value of the least one variable with a respective line in the trace file and if they are different construes the respective line in the trace file as a debugger command and invokes a debugger so as to execute the debugger command;

whereby running the program prior to carrying out said method traces one or more variables to the trace file, and step (a) includes modifying the trace file so as to replace or insert at least one traced value with the predetermined debugger command.

3. The method according to Claim 35, wherein the debugger attaches itself to a predetermined debugger command, which halts execution of the program and shows a state of the program at that time.

Sub A5 7
1 4. The method according to Claim 35, wherein the
2 program is multi-threaded and there is included the
3 further step of:

4 (c) providing a mechanism for rerunning the
5 program with identical interleaving as far
6 as instrumentation statements are
7 concerned.

1 5. The method according to Claim 1, further including
2 the step of creating the mechanism automatically using
3 the instrumentation statements.

1 6. The method according to Claim 35, wherein the
2 program includes multiple threads, each of which prints
3 an invariant associated with a status of the program,
4 said invariant having a value that remains constant
5 regardless of the interleaving.

1 7. A computer-implemented method for automatically
2 invoking a predetermined debugger function at a desired
3 location of a specific thread of a program containing at
4 least one thread, said method comprising:

5 (a) embedding within the specific thread of the
6 program at said desired location thereof a
7 utility which:

8 i) checks whether a trace file exists,

9 ii) if the trace file does not exist, creates the
10 trace file and writes a traced value of at
11 least one variable thereto at a desired
12 location of the program, and

13 iii) if the trace file does exist, compares a
14 current value of the least one variable with
15 a respective line in the trace file and if
16 they are different invokes a debugger so as
17 to execute a debugger command embedded in the
18 trace file in place of the traced value; and

Sub A57

19 (b) running the program for reading a modified trace
20 file readable by the program wherein at least one
21 traced value is replaced or augmented by said
22 debugger command.

1 8. The method according to Claim 7, wherein the
2 predetermined debugger command halts execution of the
3 program and shows the state of the program at that time.

1 9. The method according to Claim 7, wherein the
2 program is multi-threaded and there is included the
3 further step of:

4 (c) providing a mechanism for rerunning the program
5 with identical interleaving as far as instrumen-
6 tation statements are concerned.

1 10. The method according to Claim 9, including the
2 further step of creating the mechanism automatically
3 using the instrumentation statements.

1 11. The method according to Claim 7, wherein the
2 program includes multiple threads, each of which prints
3 an invariant associated with a status of the program,
4 said invariant having a value that remains constant
5 regardless of the interleaving.

1 12. The method according to Claim 7, wherein the program
2 includes multiple threads and step (a)(ii) includes:

3 i) creating for each thread a respective trace
4 file having a name which is uniquely defined
5 by a name of the respective thread;

6 thereby allowing debugger commands embedded in any
7 of the trace files to be executed during a respective one
8 of the threads.

1 13. The method according to Claim 12, further including
2 the step of automatically naming said trace files
3 according to a predetermined execution-independent naming
4 scheme.

Sub A57

1 14. The method according to Claim 13, wherein said
2 naming scheme includes the steps of:

- 3 i) assigning a root name to a root thread,
4 ii) maintaining a thread-bound index structure
5 for holding for each thread a corresponding
6 index counter which is atomically incremented
7 upon thread creation, and
8 iii) upon creation of a child thread assigning a
9 name including a prefix indicative of a name
10 of a respective parent thread and a suffix
11 indicative of an index counter of the
12 respective parent thread.

1 15. The method according to Claim 11, wherein step
2 (a) (ii) includes:

- 3 i) attempting a bipartite matching between the
4 threads and the traces such that every thread
5 has a trace which contains what the thread
6 printed, and
7 ii) if said bipartite matching is not possible,
8 then stopping the program so as to avoid
9 executing debugger commands embedded in each
10 of the traces at the wrong time.

1 16. The method according to Claim 11, further including
2 providing a mechanism for manually or automatically
3 bypassing step (a) (ii) so that traces are created in
4 respect of only a subset of the threads.

1 17. The method according to Claim 11, wherein step (b)
2 includes:

- 3 i) reading the modified trace file in respect of
4 local views of the threads only, so as to
5 avoid a need for synchronizing break-points
6 in said multiple threads.

Sub A⁵ 7

1 18. A computer program storage device readable by
2 machine, tangibly embodying a program of instructions
3 executable by the machine to perform method steps for
4 automatically invoking a predetermined debugger command
5 at a desired location of a single thread of a program
6 containing at least one thread, said method comprising:

7 (a) embedding within said program thread at said
8 desired location thereof a utility which reads a
9 trace file in which said predetermined debugger
10 command has been previously embedded; and

11 (b) running the program for reading said trace file
12 and invoking said predetermined debugger command.

1 19. A computer program product comprising a computer
2 useable medium having computer readable program code
3 embodied therein for automatically invoking a
4 predetermined debugger command at a desired location of a
5 single thread of a program containing at least one
6 thread, said computer program product comprising:

7 computer readable program code for causing the
8 computer to embed within said program thread at said
9 desired location thereof a utility which reads a trace
10 file in which said predetermined debugger command has
11 been previously embedded; and

12 computer readable program code for causing the
13 computer to run the program for reading said trace file
14 and invoking said predetermined debugger command.

1 20. A computer program storage device readable by
2 machine, tangibly embodying a program of instructions
3 executable by the machine to perform method steps for
4 automatically invoking a predetermined debugger function
5 at a desired location of a specific thread of a program
6 containing at least one thread, said method comprising:

Sub A57

(a) embedding within the specific thread of the program at said desired location thereof a utility which:

i) checks whether a trace file exists,

ii) if the trace file does not exist, creates the trace file and writes a traced value of at least one variable thereto at a desired location of the program, and

iii) if the trace file does exist, compares a current value of the least one variable with a respective line in the trace file and if they are different invokes a debugger so as to execute a debugger command embedded in the trace file in place of the traced value; and

(b) running the program for reading a modified trace file readable by the program wherein at least one traced value is replaced or augmented by said debugger command.

21. A computer program product comprising a computer useable medium having computer readable program code embodied therein for automatically invoking a predetermined debugger function at a desired location of a specific thread of a program containing at least one thread, said computer program product comprising:

computer readable program code for causing the computer to embed within the specific thread of the program at said desired location thereof a utility which:

Sub A5.7

10 computer readable program code for causing the
11 computer to check whether a trace file exists,

12 computer readable program code for causing the
13 computer to create the trace file if the trace file does
14 not exist, and to write a traced value of at least one
15 variable thereto at a desired location of the program,
16 and

17 computer readable program code for causing the
18 computer to compare a current value of the least one
19 variable with a respective line in the trace file if the
20 trace file does exist, and if they are different to
21 invoke a debugger so as to execute a debugger command
22 embedded in the trace file in place of the traced value;
23 and

24 computer readable program code for causing the
25 computer to run the program for reading a modified trace
26 file readable by the program wherein at least one traced
27 value is replaced or augmented by said debugger command.

1 22. A computer program storage device readable by
2 machine, tangibly embodying a program of instructions
3 executable by the machine to perform method steps for
4 automatically invoking a predetermined debugger function
5 at a desired location of a specific thread of a program
6 containing at least one thread, said method comprising:

7 (a) checking whether a trace file exists,

8 (b) if the trace file does not exist, creating the
9 trace file and writes a traced value of at least
10 one variable thereto at a desired location of the
11 program, and

12 (c) if the trace file does exist, comparing a current
13 value of said at least one variable with a
14 respective line in the trace file and if they are
15 different invoking a debugger so as to execute a
16 debugger command embedded in the trace file in
17 place of the traced value.

Sub A57

1 23. A computer program product comprising a computer
2 useable medium having computer readable program code
3 embodied therein for automatically invoking a
4 predetermined debugger function at a desired location of
5 a specific thread of a program containing at least one
6 thread, said computer program product comprising:

7 computer readable program code for causing the
8 computer to check whether a trace file exists,

9 computer readable program code for causing the
10 computer to create the trace file if the trace file does
11 not exist, and to write a traced value of at least one
12 variable thereto at a desired location of the program,
13 and

14 computer readable program code for causing the
15 computer to compare a current value of the at least one
16 variable with a respective line in the trace file if the
17 trace file does exist, and if they are different to
18 invoke a debugger so as to execute a debugger command
19 embedded in the trace file in place of the traced value.

1 24. A computer-implemented system for automatically
2 invoking a predetermined debugger command at a desired
3 location of a single thread of a program containing at
4 least one thread, said system comprising:

5 a code modifier for embedding within said single
6 thread at said desired location thereof a utility which
7 reads a trace file in which said predetermined debugger
8 command has been previously embedded, and

9 a processor for reading said trace file during
10 running of the program and invoking said predetermined
11 debugger command.

1 25. The system according to Claim 24, further including:

2 a file management system coupled to the processor
3 and responsive to said utility for checking whether a

Sub AS 7

4 trace file exists, and for creating the trace file if it
5 does not exist,
6 a file modifier coupled to the file management
7 system and responsive to the trace file being created for
8 writing to trace file a traced value of at least one
9 variable at said desired location of the program, and

10 a comparator coupled to the processor for comparing
11 a current value of the least one variable with a
12 respective line in the trace file and if they are
13 different construing the respective line in the trace
14 file as a debugger command and invoking a debugger so as
15 to execute the debugger command.

1 26. The system according to Claim 24, wherein the
2 debugger is adapted to attach itself to a predetermined
3 debugger command, which halts execution of the program
4 and shows a state of the program at that time.

1 27. The system according to Claim 24, wherein the
2 program is multi-threaded and there is further included a
3 replay mechanism coupled to the processor for rerunning
4 the program with identical interleaving as far as
5 instrumentation statements are concerned.

1 28. A computer-implemented system for automatically
2 invoking a predetermined debugger function at a desired
3 location of a specific thread of a program containing at
4 least one thread, said system comprising:

5 a code modifier for embedding within said single
6 thread at said desired location thereof a utility which
7 reads a trace file in which said predetermined debugger
8 command has been previously embedded,

9 a file management system coupled to the processor
10 and responsive to said utility for checking whether a
11 trace file exists, and for creating the trace file if it
12 does not exist,

Sub A57

13 a file modifier coupled to the file management
14 system and responsive to the trace file being created for
15 writing to trace file a traced value of at least one
16 variable at said desired location of the program,

17 a processor for running the program for reading a
18 modified trace file readable by the program wherein at
19 least one traced value is replaced or augmented by said
20 debugger command, and

21 a comparator coupled to the processor for comparing
22 a current value of the least one variable with a
23 respective line in the trace file and if they are
24 different construing the respective line in the trace
25 file as a debugger command and invoking a debugger so as
26 to execute the debugger command.

1 29. The system according to Claim 28, wherein the
2 program is multi-threaded and there is further included a
3 replay mechanism coupled to the processor for rerunning
4 the program with identical interleaving as far as
5 instrumentation statements are concerned.

1 30. The system according to Claim 28, wherein the
2 program includes multiple threads and the file management
3 system is adapted to create for each thread a respective
4 trace file having a name which is uniquely defined by a
5 name of the respective thread, thereby allowing debugger
6 commands embedded in any of the trace files to be
7 executed during a respective one of the threads.

1 31. The system according to Claim 30, wherein the file
2 management system is responsive to a predetermined
3 execution-independent naming scheme for automatically
4 naming said trace files.

1 32. The system according to Claim 31, wherein the file
2 management system includes:

3 an assignment unit for assigning a root name to a
4 root thread, and

Sub A57

5 a thread-bound index structure for holding for each
6 thread a corresponding index counter which is atomically
7 incremented upon thread creation;

8 said assignment unit being responsive to creation
9 of a child thread for assigning a name including a prefix
10 indicative of a name of a respective parent thread and a
11 suffix indicative of an index counter of the respective
12 parent thread.

1 33. The system according to Claim 32, wherein the
2 assignment unit is responsive to no consistent naming
3 being possible for attempting a bipartite matching
4 between the threads and the traces such that every thread
5 has a trace which contains what the thread printed, and
6 for stopping the program so as to avoid executing
7 debugger commands embedded in each of the traces at the
8 wrong time if said bipartite matching is not possible.

1 34. The system according to Claim 28, further including
2 a bypass mechanism coupled to the file modifier for
3 allowing creation of the trace file to be manually or
4 automatically bypassed so that traces are created in
5 respect of only a subset of the threads.

1 35. The system according to Claim 28, wherein the
2 processor is adapted to read the modified trace file in
3 respect of local views of the threads only, so as to
4 avoid a need for synchronizing break-points in said
5 multiple threads.

IS 999-010

23